



Eur päisches
Patentamt

Eur pean
Patent Office

Office européen
des brevets

PCT/GB 99 / 0 3 6 0 6

29 FEBRUARY 2000

ETU

REC'D 29 FEB 2000

WIPO

PCT

Bescheinigung

Certificate

Attestation

09/806694

Die angehefteten Unterla-
gen stimmen mit der
ursprünglich eingereichten
Fassung der auf dem näch-
sten Blatt bezeichneten
europäischen Patentanmel-
dung überein.

The attached documents
are exact copies of the
European patent application
described on the following
page, as originally filed.

Les documents fixés à
cette attestation sont
conformes à la version
initialement déposée de
la demande de brevet
européen spécifiée à la
page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

99306394.0

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,
LA HAYE, LE

24/02/00

THIS PAGE BLANK (USPTO)



**Eur päisches
Patentamt**

**Eur pean
Patent Office**

**Office eur péen
des brevets**

**Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation**

Anmeldung Nr.:
Application no.: 99306394.0
Demande n°:

Anmeldetag:
Date of filing: 13/08/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
BRITISH TELECOMMUNICATIONS public limited company
London EC1A 7AJ
UNITED KINGDOM

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
A co-ordinating system

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragsstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du dépôt:

Bemerkungen:
Remarks:
Remarques:

THIS PAGE BLANK (USPTO)

A CO-ORDINATING SYSTEM

The present invention relates to a co-ordinating system, suitable particularly but not exclusively for co-ordinating autonomous intelligent agents.

- 5 An autonomous intelligent agent is a computer program that acts for an entity such as a user, a piece of equipment or a business to assist in dealing with information management tasks. The intelligent agent usually holds data in relation to the entity that it represents, has a set of constraints or conditions to determine its behaviour and, most importantly, is provided with decision making software for
- 10 making decisions on behalf of the entity within, or as a result of, the constraints and conditions. Agents generally act within a system and the decisions that an agent makes can result in activity by the system. Each agent typically has its own interface and interacts with the entity in its own particular way, and the agents may communicate with one another by message passing (such as the Open Messaging
- 15 Architecture of Zeus, Nwana, Ndumu and Lee¹).

- Various workers have been concerned with the development of systems that incorporate intelligent agents to provide more coherent information to users, for applications ranging from robotic to internet applications, and by addressing aspects such as agent architecture, collaboration, autonomy and organisation² and agent
- 20 mobility. Numerous agent architectures have been proposed in the literature, addressing key features that an agent should have, and these may be broadly categorised as one of the following types of agents: reactive, deliberative and interacting agents³. A well known deliberative agent architecture includes the Belief-Desire-Intention architecture (BDI), which has three types of elements for performing
- 25 basic roles of analogous types in humans: beliefs, desires and intentions. These represent an agent's information and knowledge, preferences, and goals respectively, and each goal is assumed to be fulfilled by a sequence of actions (such sequences of

¹ 'ZEUS: An advanced tool-kit for engineering distributed multi-agent systems', Proceedings of the third International Conference on Practical applications of intelligent agents and multi-agent technology, 1998, 377-391

² Kautz, Selman, Coen, 'Bottom-up design of software agents' Communications of the ACM 37(7) 1994

³ J. P. Muller, 'Control Architectures for Autonomous and Interacting Agents: A Survey'

actions are referred to as 'plans'). This architecture has been implemented in the Procedural Reasoning System (PRS)⁴ system, which provides a system comprising a plurality of agents that is based on a single thread of execution, includes records of current work plans, and does not attempt to schedule work ahead of the current
5 time. The PRS system thus results in an inability to reschedule tasks or revise intentions in response to changes in the system environment, or to schedule intentions at specific points in time in the future.

Patent application W099/05597, in the name of the present applicant, describes a software building environment for building a software system using
10 collaborative agents ('ZEUS '). A system built by using the environment can then be used for instance in control, monitoring and/or management of a process or apparatus. The software system, as built, comprises at least one software module, and comprises a collaborative agent, which gives the system access to at least one collaboration or co-ordination strategy, expressed for instance as a rule or algorithm.
15 Each collaboration agent includes a planning/scheduling module that comprises a commitment table detailing scheduled times for executing future tasks, and in particular this offers an overbooking facility to ensure maximised use of resources. Thus this agent architecture is capable of scheduling and rescheduling intentions at specific points in time in the future.

20

According to the present invention there is provided apparatus for co-ordinating tasks to be executed by a computer system, including
(i) a world model comprising a list of tasks, wherein each task has a timeslot associated therewith,
25 (ii) scheduling software to schedule the tasks, and
(iii) execution software to effect execution of a scheduled task,
which apparatus is operable to receive and process task information such that when the task information includes a request to supply information, the scheduling software schedules execution of the request to occur in a free timeslot of the world
30 model.

⁴ Rao, Georgeff 'Modelling rational agents within a BDI architecture' Technical Report 14, AUS AI Institut , Carlton, AUS 1991

According to a further aspect of the present invention there is provided a system for assisting in the management of information flows for a user, including apparatus for co-ordinating tasks to be executed by a computer system, software
5 means operable to concurrently execute a plurality of processes and a plurality of intelligent autonomous systems that help the user with certain computer based tasks and whose interaction with the user is controlled by the apparatus for co-ordinating tasks.

10 Further aspects, features and advantages of apparatus for co-ordinating tasks will now be described, by way of example only, as an embodiment of the present invention, with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of the hardware of a computer system configured to run apparatus for co-ordinating tasks of the present invention,

15 Figure 2 is a schematic diagram showing apparatus for co-ordinating tasks interacting with a plurality of intelligent agents according to the present invention,

Figure 3 is a schematic diagram showing apparatus for co-ordinating tasks of Figure 2, and

Figure 4 is a schematic diagram showing the Java thread processes
20 comprising part of the apparatus of Figure 3.

In the following, the terms "goal", and "task" are used. These are defined as follows:

A "goal": a logical description of a resource (fact) which it is intended an agent should produce;

25 A "task": an action or an event that a system or agent performs at a specific time.

Figure 1 shows a generally conventional computer system 100 that comprises: a conventional keyboard 101; a display screen 103, such as a CRT or plasma screen; a mouse 105; a processor 107 such as a Pentium™ processor;
30 random access memory 109; a hard disc drive 111; an audio input 113 such as a microphone to detect utterances from the user; and input/output interfaces 115 to connect the workstation to a local area network (LAN) and wider area networks

(WAN) such as the internet, to facilitate data exchange including e-mail messaging with remote users connected to such networks. The interface 115 also allows control of a plain old telephone set (POTS) and the components shown in Figure 1 are interconnected by a common bus 117. In addition to the single system configuration shown in Figure 1, several computer systems (not shown) may be interconnected over a local area network via the input/output interface 115.

Referring also to Figure 2, apparatus 305 for co-ordinating tasks to be executed by a computer system 100 may be stored on the hard disc drive 111 for processing by the processor 107. The apparatus 305 may be part of an intelligent assistant system 219 which enables users to devote their time to highly complex tasks while the system 219 takes some decisions on behalf of the user based on previous observations of the user, thus enabling the user to increase productivity. Typical tasks to be performed by the system include time, information and communication management. When the computer system comprises several computer workstations, several intelligent assistant systems 219 may be active and may communicate with one another.

As shown in Figure 2, such a system 219 may additionally comprise a set of autonomous systems 201, 203, 205, 207, 209, 211, generally referred to as agents, specialising in various tasks such as diary management, telephone call filtering and e-mail prioritisation, web search and telephone directory enquiry. Each agent has its own interface and interacts with the user in its own particular way, and the agents communicate with one another by message passing. These agents are essentially reactive agents that respond to events in the user's environment (such as e-mails and telephone calls) by initiating interactions with the user.

The inter-agent collaboration of known systems described above is provided by the agents 201, 203, 205, 207, 209, 211 of the present system 219 (albeit that the implementation of collaboration is different) but the system 219 additionally provides co-ordination of the presentation of information to the user. In general, after an agent has completed its task, its next course of action will be to present the corresponding task results to the user. However, in the absence of some means of controlling when these results are presented, the user may be overloaded with information from many agents simultaneously. Thus co-ordination of agent

information is necessary to avoid increasing the cognitive load on the user. This co-ordination typically includes managing the interaction between the agents and the user; performing tasks on behalf of the user that requires the action of more than one agent; and scheduling actions to be performed at appropriate times. This therefore
5 involves receiving task information from other agents, processing the task information into executable system actions, such as the action of allowing an agent to display information to the user, and maintaining a temporal database of these system actions. Although the system disclosed in patent application W099/05597 (described above) is able to schedule and re-schedule tasks, the issue of scheduling
10 presentation of task information to the user so as to avoid an information overload from many agents at any one time is not addressed.

The apparatus 305 of the present invention may be a co-ordinating agent having a three-tier architecture (reactive - deliberative - meta-reasoning) capable of scheduling and subsequently executing its own future actions, and it is distinguished
15 from the other reactive agents 201, 203, 205, 207, 209 described above by virtue of this scheduling capability. Thus the co-ordinating agent, hereinafter referred to as a 'co-ordinator', is generally descriptive of the apparatus 305 as an embodiment of the present invention.

The co-ordinator 305 functions under the control of software means 403,
20 which may be provided by Java threads, although it is understood that that the use of Java is inessential to the invention and that any other method of concurrently running multiple processes would provide the software means 403. Java is a good choice of language for developing multi-agent applications because of its object-oriented and multi-threaded characteristics, enabling each agent to comprise many
25 objects and several threads. It also has the advantage of being portable across operating systems, as well as providing a rich set of class libraries that include excellent network communication facilities.

Thus referring to Figure 3, the present invention provides a co-ordinator 305
30 for co-ordinating tasks to be executed by the computer system 100, including scheduling software 307 to schedule and/or reschedule tasks and execution software 309 to effect execution of the same. The co-ordinator 305 is operable to receive task

information 311 and to maintain a temporal record 315 of the schedule of the tasks, such that when a change is made to a task or a new task is sent to the co-ordinator 305, it 305 informs the scheduling software 307 and updates the temporal record 315 accordingly. The co-ordinator 305 also includes a library 323 of task plans, each
5 of which task plans includes a pre-specified action list actionable to perform a corresponding task. The user can explicitly specify an interruption status for allowing or otherwise interruptions to the user, and this may be input to the co-ordinator as task information 311. Typically, the interruption status includes information as to whether the user will accept or refuse interruptions (from, for example telephone
10 calls and e-mail notifications) and the co-ordinator 305 includes a simple interface allowing the user to set these preferences.

The co-ordinator 305 also includes a world model 313, which world model 313 comprises a diary of user tasks and the interruption status, and is accessed by the co-ordinator 305 when scheduling tasks and updating the temporal record 315.
15 When the co-ordinator 305 forms part of the intelligent assistant system 219, which 219 includes a diary assistant 211 as one of the intelligent agents, the world model 313 replicates the information stored by the diary assistant 211. The world model 313 may store the diary of user tasks as a list of tasks, and each task has a timeslot associated therewith. Each timeslot is defined by a start time and a duration, and
20 when the task information 311 includes a request to supply information to the user, the scheduling software 307 schedules execution of the request to occur in a free timeslot of the world model 313. If the request involves a task to be performed by the co-ordinator 305 and/or other agents, the timeslot duration may be implicit in the task when scheduled, and this may be used to update the temporal record 315,
25 together with the time that the task was initiated. In practice, if the request is to display information to the user and if the current timeslot is not free, the co-ordinator 305 will either not attempt to schedule the task, and refuse the request, or will schedule the task for a free timeslot in the future, having regard to the world model 313, and the corresponding executable task will be output from the co-ordinator 305
30 at that time. In the first situation the source of the request will have to re-initiate the request at a later time. In the second situation and the first situation when the current timeslot is free, the corresponding executable task may simply be

communication between the co-ordinator 305 and the source of the request to permit the source to communicate with the user.

Java threads 417, 419, 421, shown in Figure 4, are operable to concurrently execute a plurality of processes, each of which is responsible for executing various processes comprising the co-ordinator 305. When the co-ordinator 305 receives task information 311, a first Java thread 417 translates this into a system goal 427, and retrieves a corresponding plan 429 from the plan library 323. This plan 429 consists of a set of action templates for achieving a particular goal, and the thread 417 is required to instantiate 431 the action template in order to form an executable task using parameters supplied with the goal 427. The task information 311 may include parameters such as a deadline time, which is transferred to the system goal 427, and is then used in conjunction with the action template to specify execution times and a corresponding executable task 433 by the scheduling software 307.

Two types of deadlines may be specified: "as soon as possible", and "as late as possible before the deadline". The goal 427 may have been filtered by a filter (not shown), which is specified by a set of meta-rules and whose job is to select only those tasks that are desirable and are believed to be achievable by the co-ordinator 305. The scheduling software 307 schedules tasks based on the preferred type of scheduling ("as soon as possible", or "as late as possible before the deadline"), taking into account, by consulting the world model 313, other tasks to be executed. Once an action has been scheduled, the temporal record 315, which is a list of active tasks to be performed by the whole system 219, is updated.

Once a task has been processed into an executable task 433 as described above, a second thread 419 picks up the task. The second Java thread 419 repeatedly checks 435 to see whether a task needs executing, and if there is a task to be executed, the execution software 309 passes the information required to execute the task from the co-ordinator 305, as shown in Figures 3 and 4. The second thread 419 may run every 15 seconds to check whether tasks need executing, and is capable of retrieving one task in each cycle. If a series of tasks have been retrieved in successive cycles, the tasks are executed sequentially depending on the order in which they were retrieved.

A third Java thread 421 processes the updating of new task information 311, which includes maintaining the world record 313 of the user's activities. This is achieved in part by checking every second for new task information 311, which includes checking for the interruption status information, and by interfacing with the first and second threads 417, 419. New task information 311 may effect deletion of any goals no longer required (corresponding to new information task 311) and scheduling of a replacement (if applicable), or of a new goal. This is shown in Figure 4 as a feedback path 439, containing task information 311, to the first thread 417.

The co-ordinator 305 may also include user preference software 303, which in the case of the co-ordinator 305 may include preferred times for performing various tasks, thus enabling the co-ordinator 305 to schedule actions in response to various system goals (such as issuing reminders to read low priority e-mails). In terms of the standard three-tier agent architecture described above, the task information 311 forms part of the reactive layer; the world model 313, scheduling software 307, goals 427, plan library 323 and execution software 309 form part of the deliberative layer; and the user preference software 303 and temporal record 315 form part of the meta-reasoning layer.

As described above, the co-ordinator 305 is also operable to manage interactions between the user and the intelligent assistant system 219. Thus, when the system 219 includes a plurality of intelligent agents 201, 203, 205, 207, 209, 211, the co-ordinator 305 interacts with the intelligent agents so as to schedule presentation of their information to the user, taking into account the interruption status, such as "will not accept any interruptions". This is shown schematically in Figure 2, and the intelligent agents may include at least some of a diary assistant, an e-mail assistant, a Web assistant and a yellow pages assistant. When the system 219 includes the diary assistant 211, this is used to re-set the interruption status every 30 minutes, so that if the user has forgotten to re-set the status to "active" (i.e. "will accept interruptions"), the system 219 takes control and allows interruptions from events such as meeting reminders etc. Clearly the user can override this automatic switch if desired.

As can be appreciated from the foregoing description, the co-ordinator 305 is not a centralised controller for the system 219. Although the co-ordinator 305 can

request the agents 201, 203, 205, 207, 209, 211 to effect execution 309 of the tasks passed from the co-ordinator 305, the agents may not perform these tasks. The agents communicate with each other using the Zeus Open Messaging Architecture (detailed in footnote 1), and they 201, 203, 205, 207, 209, 211
5 operate mostly under the operation of the user, although as described above, notification of information to the user is routed through the co-ordinator 305 as shown in Figure 2 by the squares 241. In Figure 2, the ellipses 243 represent requests from the co-ordinator 305 to the agents, and may correspond to the information for effecting task execution 309. The diamonds 245 represent a record
10 of the user's preferences or interests, having been extracted from a database 247 containing user profile data. Thus the agents 201, 203 linked to the diamonds 245 may be web and yellow pages assistants.

The following scenario, of a lunch booking entered into the diary by the user,
15 together with a non-interruptible one-hour meeting starting at noon, illustrates the present invention in operation:

The diary agent, forming one of the intelligent agents, sends a message to the co-ordinator 305, represented as a new information task 311 in Figure 3, describing the lunch booking and meeting. The information task 311 includes parameters relevant to
20 the same, which for this case include a deadline of 5 minutes before the end of the meeting, and details of the person with whom he would be lunching. This 311 is translated into a goal 427, and the appropriate plan 429 is retrieved from the plan library 323. A typical plan 429 for such a scenario may include a yellow pages search for a restaurant; finding the web page of the person having lunch with the user; and
25 reminding the user of the lunch appointment. The scheduling software 307 then specifies a corresponding executable task 433, which may be passed to the execution software 309, and includes an action on the co-ordinator 305 to cause the appropriate agents to process their respective actions (in this case via ellipses 243 to web and yellow pages assistants 201, 203). The following code fragments describe
30 these processes:

{

10

```

//basic goal reduction planner
Goal = GoalList.getGoal ();
If (Goal != NULL)
{
5       Plan = PlanLibrary.fetchPlan(Goal);
       If (Plan != NULL)
           Scheduler.schedule (Goal, Plan);
           Goal.setStatus (REDUCED);
       }
10 }
{
//assumes that now is the current time
Task = TaskList.getTask (now);
While (Task != NULL)
15 {
       executeTask (Task);
       Task.setStatus (COMPLETED);
       Task = TaskList.getTask (now);
       }
20 }

```

When the agents 201, 203 have finished searching, they have to notify the user of the results. This is routed via the co-ordinator 305, shown by squares 41 on Figure 3, in order to determine whether the user is able to accept interrupts. The co-ordinator 305 accesses the world model 313, which includes details of all of the user's current and future tasks and the interruption status, in order to anticipate a next available timeslot to interrupt the user with the information. Once a timeslot has been ascertained, the co-ordinator 305 may pass an information task 311, including the ascertained time as a parameter, to the scheduling software 307 and this is scheduled in with other user's tasks in the manner described above (goal 427 – plan

429 - instantiate 431 and schedule 433 - send to execute 309 by another agent if required).

The present invention thus extends existing agent systems, such as the PRS
5 ZEUS systems described above, to include temporally specific goals and intentions,
and specifically includes a co-ordinator for scheduling and co-ordinating the
presentation of information from system agents to the user. The co-ordinator
maintains a world model of the system tasks and uses this for both scheduling
interruptions to the user and for rescheduling actions in responses to changes in the
10 system tasks.

THIS PAGE BLANK (USPTO)

CLAIMS

1. Apparatus for co-ordinating tasks to be executed by a computer system, including
- 5 (i) a world model comprising a list of tasks, wherein each task has a timeslot associated therewith,
- (ii) scheduling software to schedule the tasks, and
- (iv) execution software to effect execution of a scheduled task,
- which apparatus is operable to receive and process task information such that when
- 10 the task information includes a request to supply information, the scheduling software schedules execution of the request to occur in a free timeslot of the world model.
2. Apparatus according to claim 1, wherein each timeslot is defined by a start
- 15 time and a duration.
3. Apparatus according to claim 1 or claim 2, wherein the free timeslot is the next available timeslot.
4. Apparatus according to any of the preceding claims, wherein the request to
- 20 supply information is a request to supply information to the user.
5. Apparatus according to any one of the preceding claims, including a library of task plans, each of which task plans includes an action list actionable to perform a
- 25 corresponding task.
6. Apparatus according any one of the preceding claims, wherein the user can explicitly specify an interruption status for allowing or otherwise interruptions to the user.
- 30 7. Apparatus according to claim 6, wherein the world model includes the interruption status.

8. Apparatus according to claim 1 or claim 7, including user preference software, which user preference software details preferred actions of the user relating to task information.

5

9. A system for assisting in the management of information flows for a user, including apparatus according to any one of claims 1 to 8, and software means operable to concurrently execute a plurality of processes.

10 10. A system according to claim 9, wherein the software means includes a plurality of Java threads.

11. A system according to anyone of the preceding claims, including a plurality of intelligent autonomous systems that help the user with certain computer based tasks.

15

12. A system according to claim 11, wherein the apparatus interacts with the intelligent autonomous systems to schedule presentation of information to the user.

13. A system according to claim 12, wherein the intelligent autonomous systems
20 include at least some of a diary assistant, and e-mail assistant, a telephone assistant and a web assistant.

14. A system according to claim 13, wherein the world model replicates a log of user events maintained by the diary assistant.

25

15. A computer program product including a storage medium in which is stored computer readable code corresponding to a system as defined in any preceding claim.

ABSTRACT

Apparatus (219) for co-ordinating tasks to be executed by a computer system (100), including

- 5 (i) a world model (313) comprising a list of tasks, wherein each task has a timeslot associated therewith,
 - (ii) scheduling software (307) to schedule the tasks, and
 - (v) execution software (309) to effect execution of a scheduled task,
- which apparatus (219) is operable to receive and process task information (311) such
- 10 that when the task information (311) includes a request to supply information, the scheduling software (307) schedules execution of the request to occur in a free timeslot of the world model (313).

Figure (3)

THIS PAGE BLANK (USPTO)

1/4

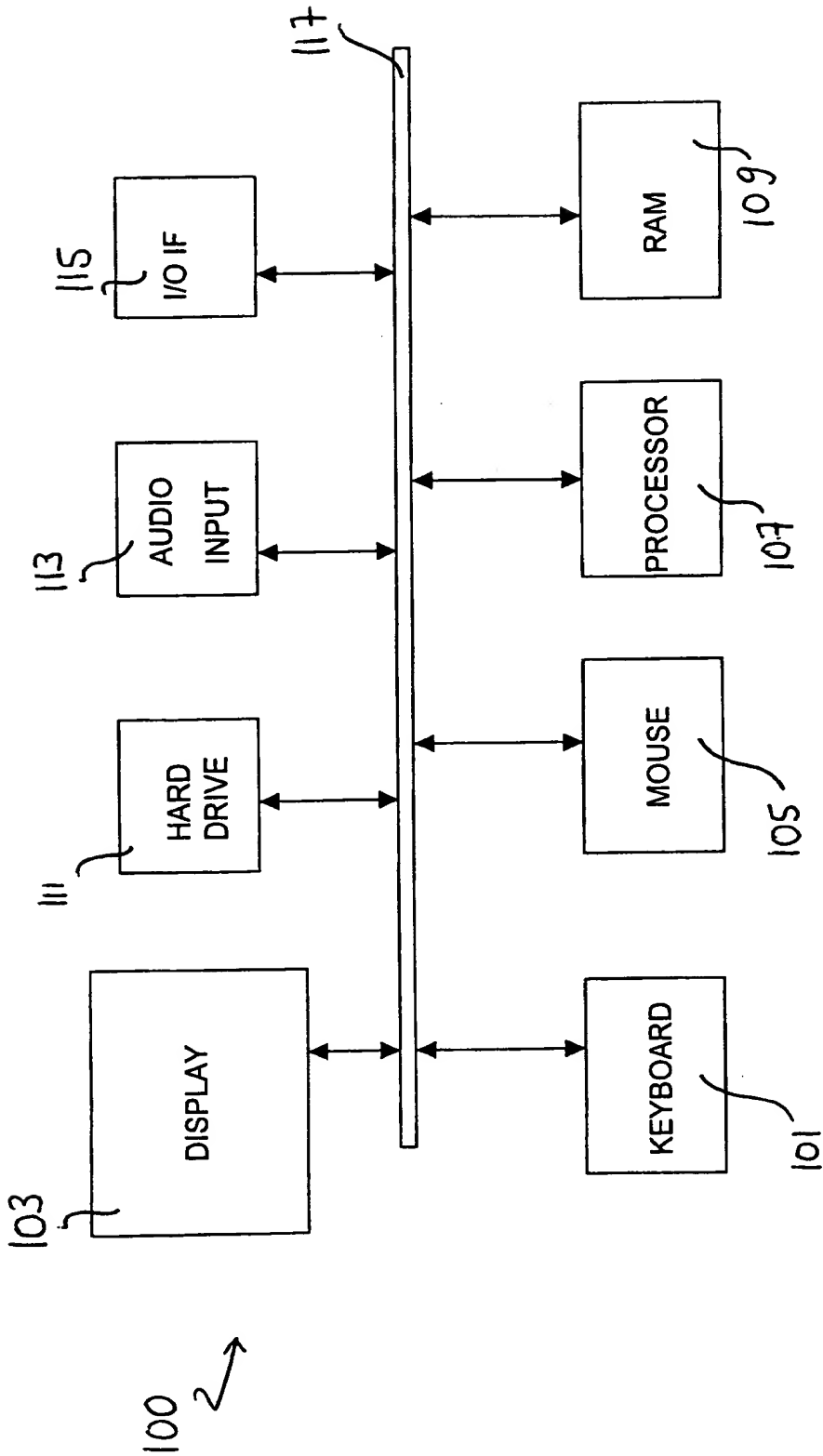


FIG 1.

2/4

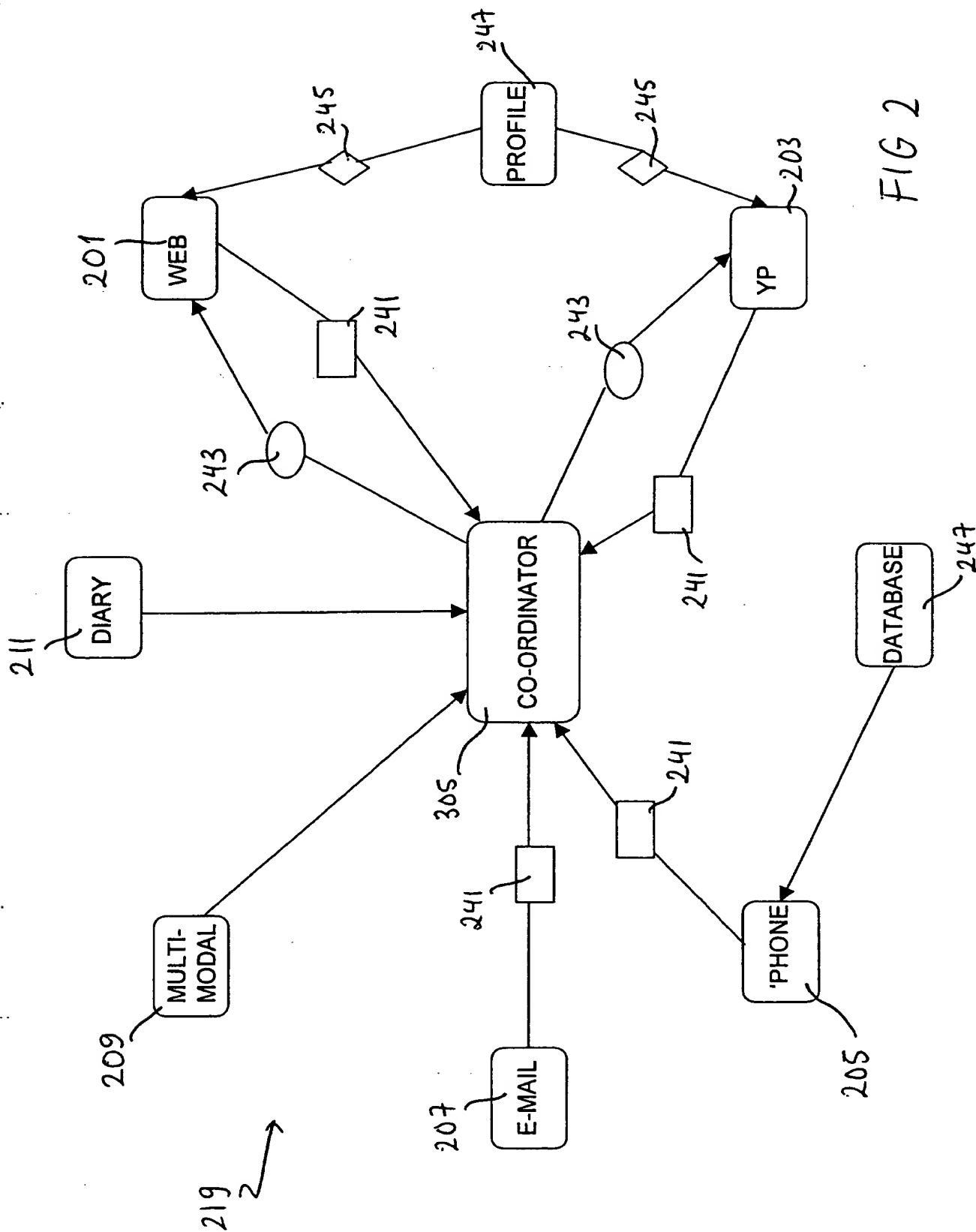


FIG 2

3/4

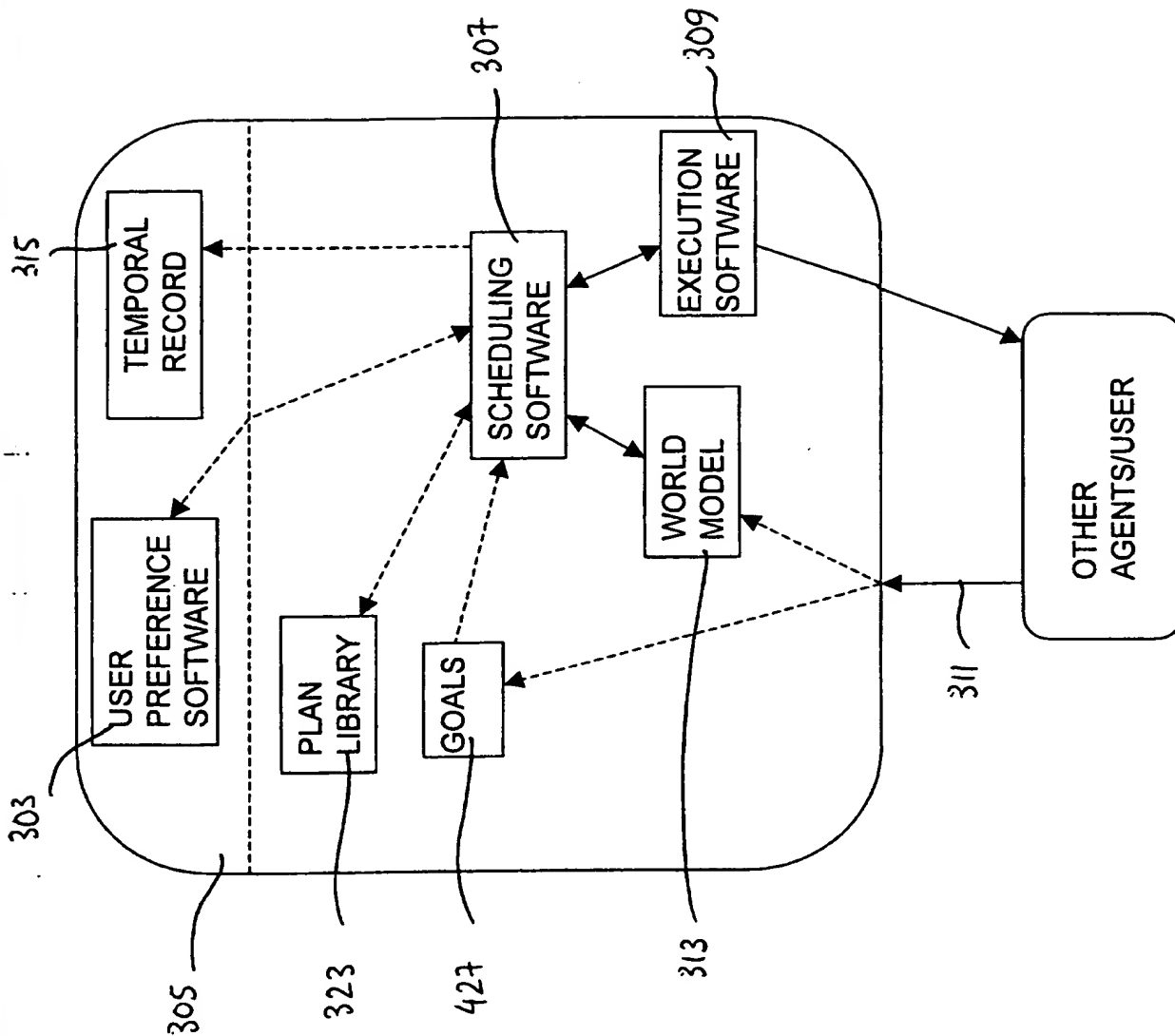


FIG 3

4/4

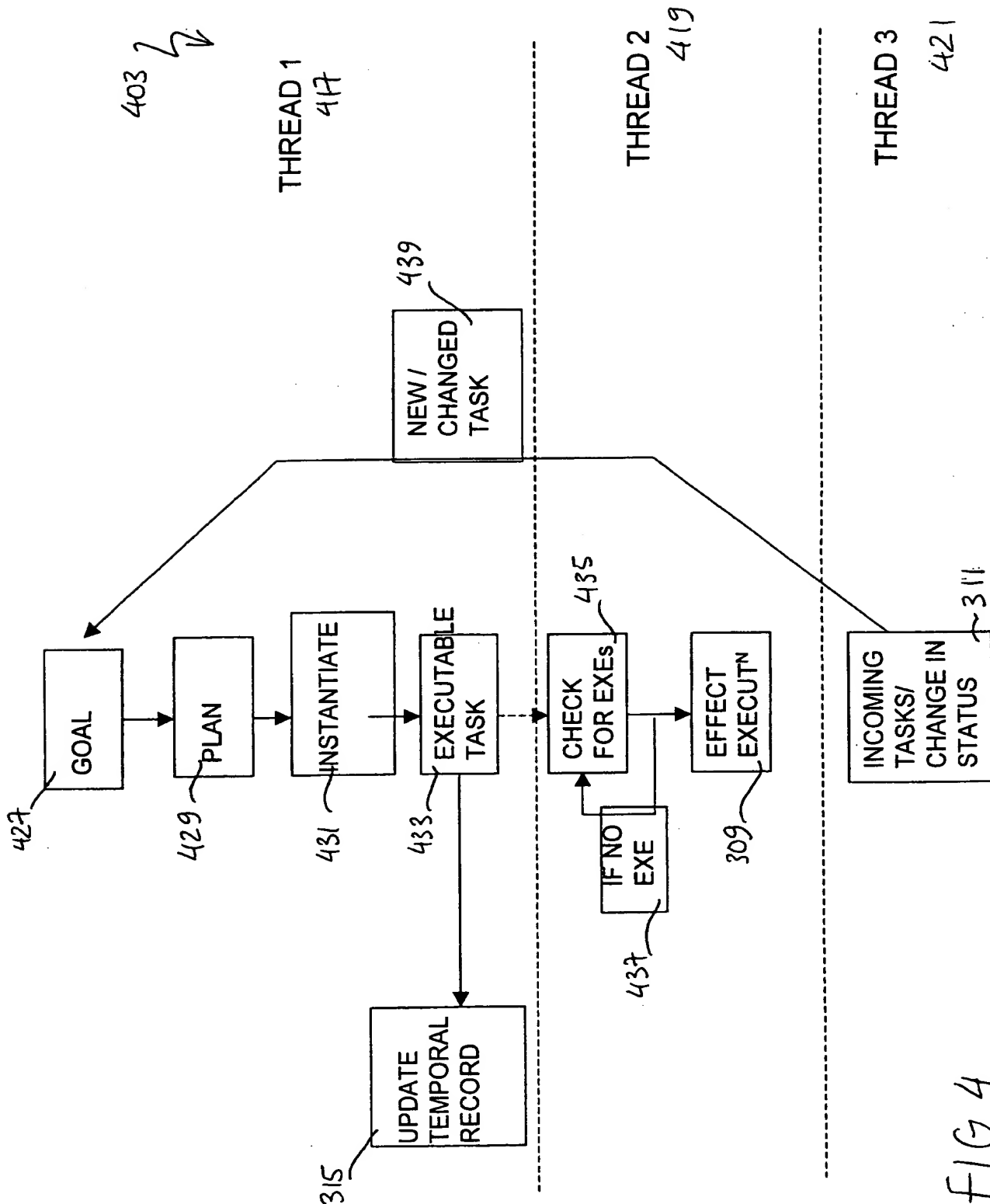


FIG 4